Pabna University of Science and Technology

Department of

Computer Science and Engineering

Faculty of

Engineering and Technology

Assignment On

Course Code: CSE 3208

Course Title: Digital Signal Processing Sessional

Submitted by:	Submitted to:
Name: Md. Habibul Islam	Dr. Md. Imran Hossain
Roll: 200124	Associate Professor.
Session 2019-20.	Dept. Of ICE,
3rd year 2nd semester	Pabna University of Science and Technology
Dept. Of CSE, PUST.	Pabna, Bangladesh.

Date of Submission: June 07,2024

Index

Problems Link: https://github.com/habibulislamruddro/DSP-Final-Assignment

Lab Question 8: Design an FIR filter to meet the following specifications—Passsband edge=2KHz, Stopband edge= 5KHZ, Fs=20KHz, Filter length =21, use Hanning window in the design.

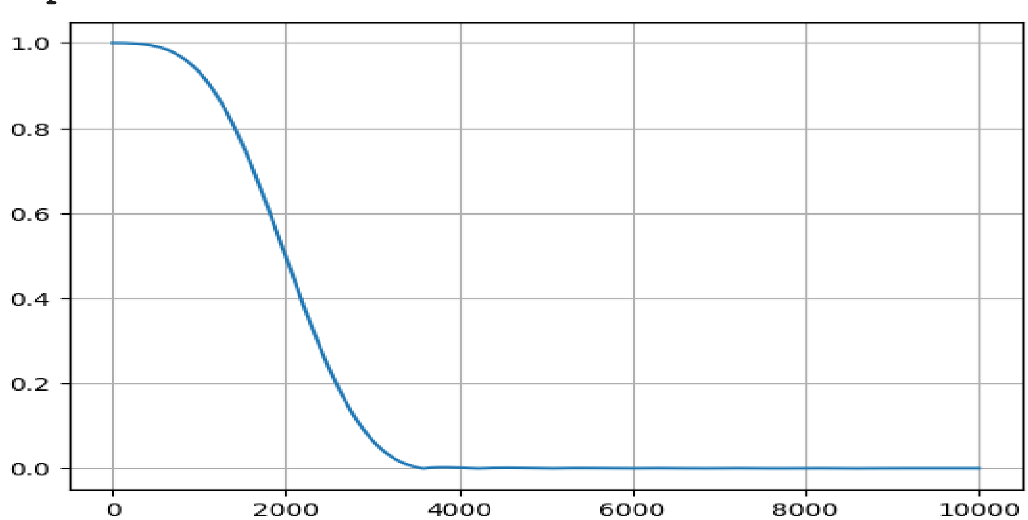
Code:

```
import numpy as np
from scipy.signal import firwin,freqz
import matplotlib.pyplot as plt


pass_band = int(input("Enter Pass Band:"))
stop = int(input("Enter stop Band:"))
fs1 = int(input("Enter Fs:"))
filter_length = int(input("Enter Filter length:"))

tap = firwin(filter_length, pass_band,width=stop-
pass_band,pass_zero='lowpass',fs=fs1,window='hann')

response = freqz(tap, worN=8000)


plt.plot(fs1*.5*response[0]/(np.pi), np.abs(response[1]))
plt.grid()
plt.show()
```

Input:

```
Enter Pass Band:2000
Enter stop Band:5000
Enter Fs:20000
Enter Filter length:21
```

Output:



Lab Question 9: Creating a signals ‘s‘ with three sinusoidal components (at 5,15,30 Hz) and a time vector ‘t‘ of 100 samples with a sampling rate of 100 Hz, and displaying it in the time domain. Design an IIR filter to suppress frequencies of 5 Hz and 30 Hz from given signal.

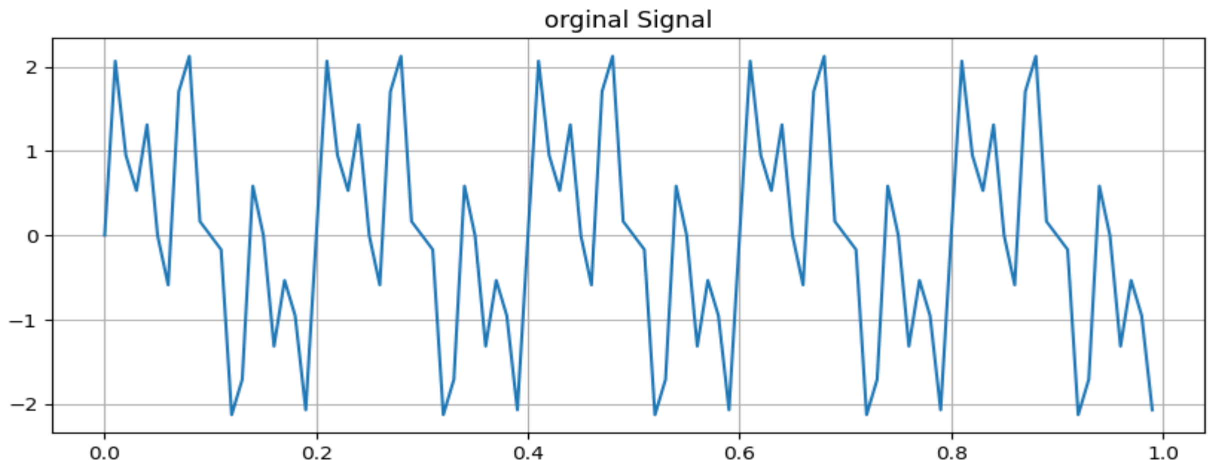
Code:

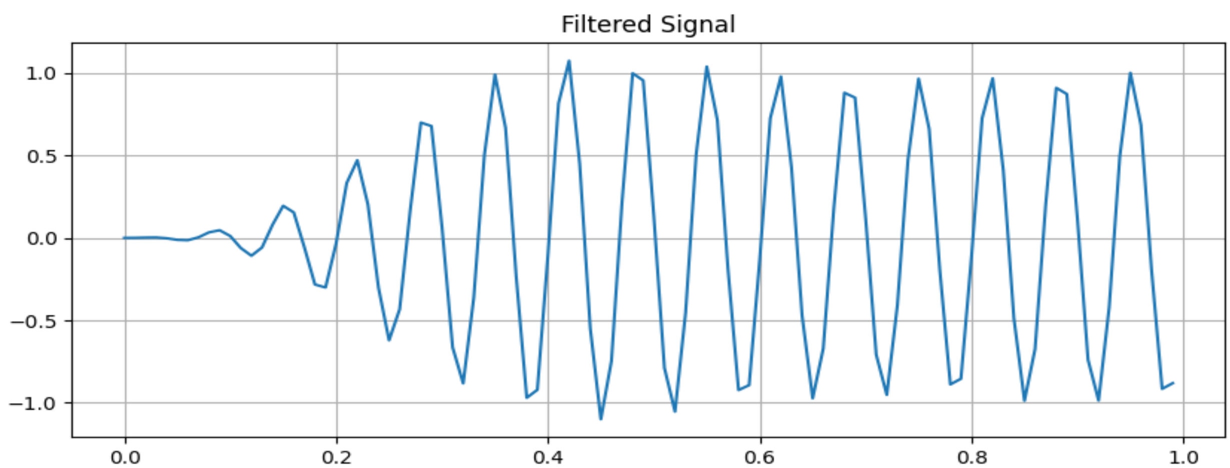
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import butter,lfilter,freqz
fs = int(input("Enter the sampling rate in Hz: "))
f1 = int(input("Enter first frequency in Hz: "))
f2 = int(input("Enter second frequency in Hz:"))
f3 = int(input("Enter third frequency in Hz:"))
time = np.arange(0,1,1/fs)
s=
np.sin(2*np.pi*f1*time)+np.sin(2*np.pi*f2*time)+np.sin(2*np.pi*f3*time
)
def butter_stopband_filter(data,lowcut,highcut,fs,order=4):
    nyquist= 0.5*fs
    low = lowcut/nyquist
    high = highcut/nyquist
    b,a = butter(order,[low,high],btype='bandpass')
    y= lfilter(b,a,data )
    return y
plt.figure(figsize=(10,4))
plt.plot(time,s)
plt.title('orginal Signal')
plt.grid()
plt.show()
filtered_signal= butter_stopband_filter(s,13,17,fs )
plt.figure(figsize=(10,4))
plt.plot(time,filtered_signal)
plt.title('Filtered Signal')
plt.grid()
plt.show()
plt.show()
#fft or dft calculation
X_mag = np.fft.fft(filtered_signal)
X_freq = np.fft.fftfreq(len(X_mag), 1/fs)
magnitude = np.abs(X_mag)/len(X_mag)

plt.figure(figsize=(10,4))
plt.stem(X_freq,magnitude)
plt.title('magnitude')
plt.grid()
plt.show()
```

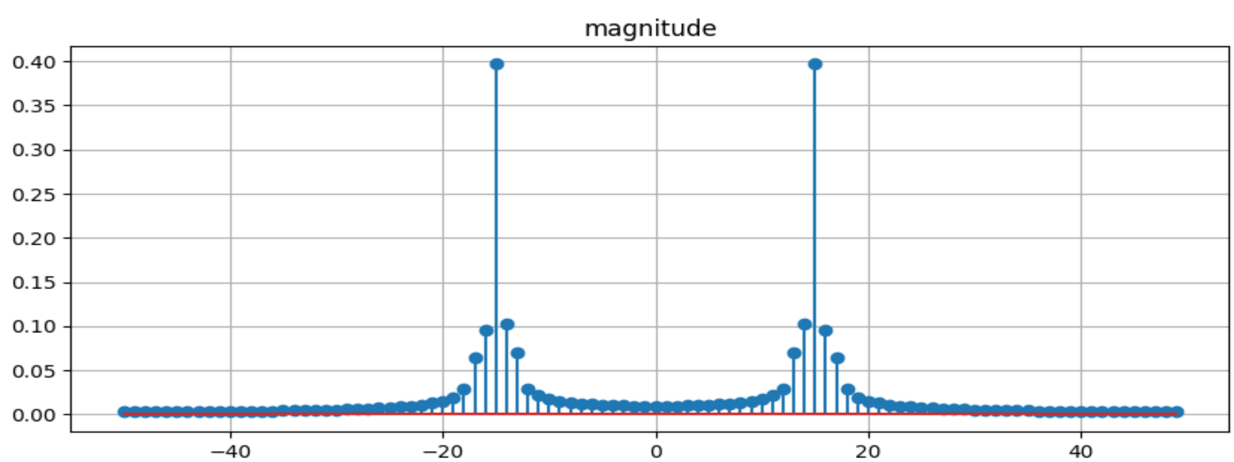
Input:

```
Enter the sampling rate in Hz: 100
Enter first frequency in Hz: 5
Enter second frequency in Hz:15
Enter third frequency in Hz:30
```

Output:







Lab Question 10: Design a Lowpass filter to meet the following specifications—Passsband edge=1.5KHz, Transition width = 0.5KHz, Fs=10KHz Filter length =67; use Blackman window in the design.

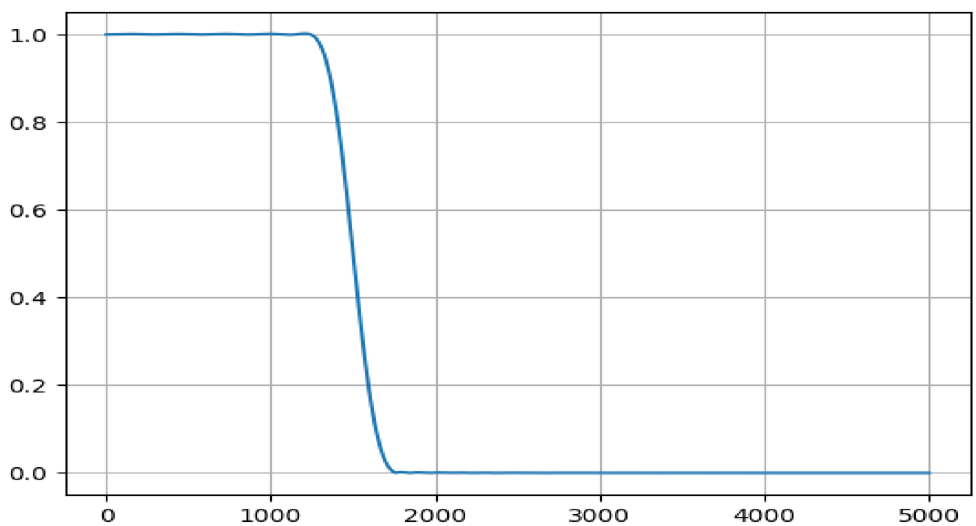
Code:

```
import numpy as np
from scipy.signal import firwin,freqz
import matplotlib.pyplot as plt
import scipy.signal as signal

pass_band = int(input("Enter Pass Band: "))
tran_width = int(input("Enter Transition Width: "))
fs = int(input("Enter Fs: "))
filter_length = int(input("Enter Filter Length: "))
taps =
signal.firwin(67,pass_band,fs=fs,pass_zero='lowpass',window='
blackman',width=tran_width)
response = signal.freqz(taps,worN=8000)


plt.plot(fs*response[0]/2/np.pi,np.abs(response[1]))
plt.grid()
```

Input:

```
Enter Pass Band: 1500
Enter Transition Width: 500
Enter Fs: 10000
Enter Filter Length: 67
```

Output:

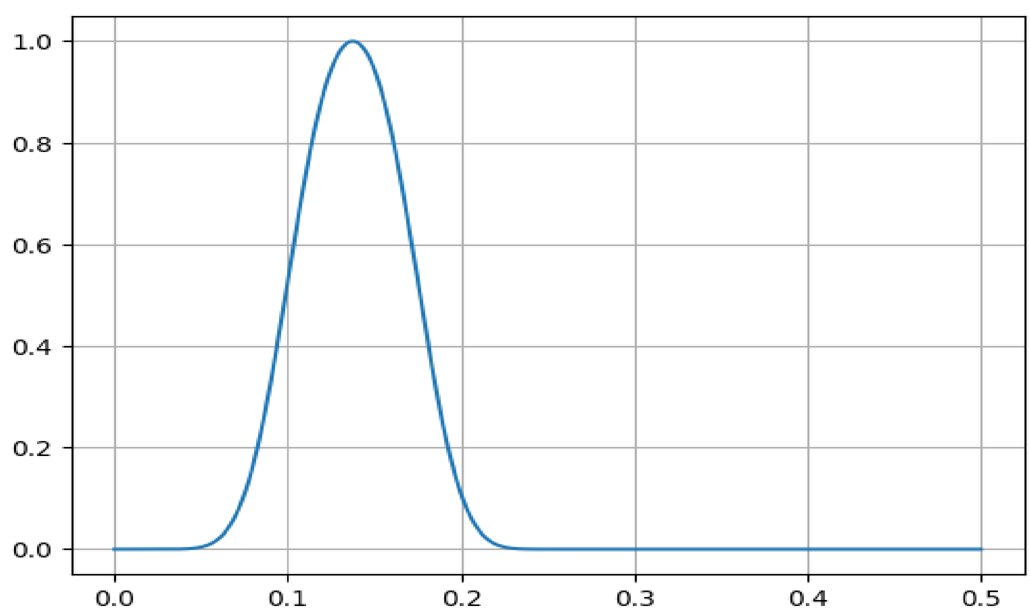


Lab Question 11: Design a bandpass filter of length M=32 with passband edge frequencies fp1=0.2 and fp2=0.35 and stopband edge frequencies fs1=.1 and fs2=0.425.

Code:

```
import matplotlib.pyplot as plt
import numpy as np
import scipy.signal as signal
pass_band = [.2,.35]
tran_width = .425-.1

filter_length = 67
taps =
signal.firwin(filter_length,pass_band,pass_zero='bandpass',wind
ow='blackman',width=tran_width)
response = signal.freqz(taps,worN=8000)
taps =
signal.firwin(67,pass_band,pass_zero='bandpass',window='blackma
n',width=tran_width)
response = signal.freqz(taps,worN=8000)


plt.plot(response[0]/2/np.pi,np.abs(response[1]))
plt.grid()
```

Output:



Assignment Question 2: Highpass Filter

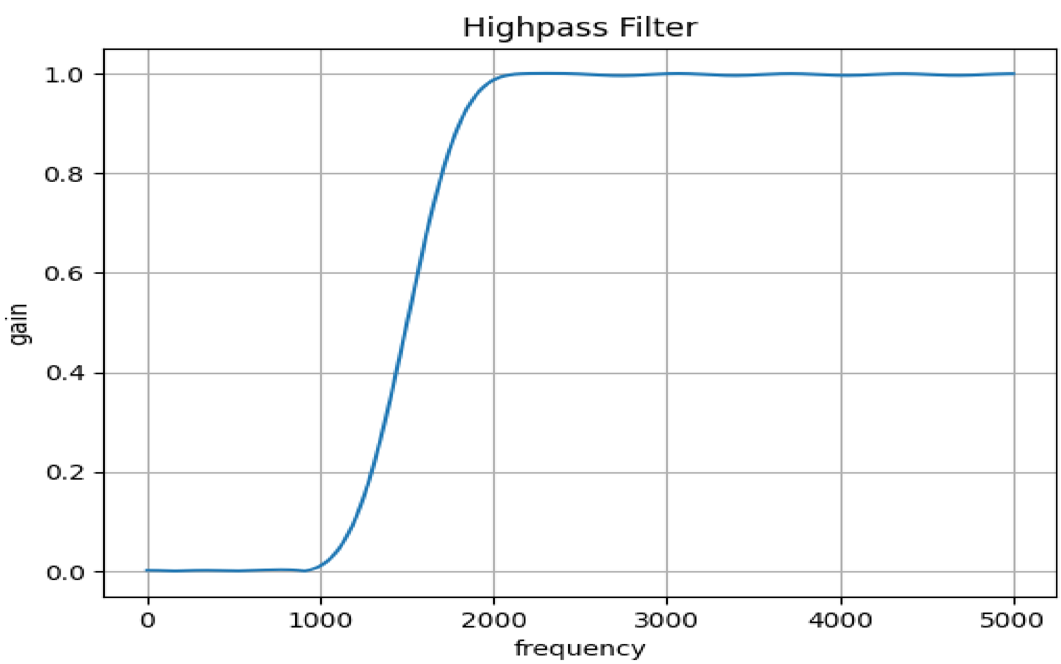
Code:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as sig
#parameters
pass_bands = int(input("Enter the pass band: "))
stop_bands = int(input("Enter the stop band: "))
fs = int(input("Enter the sampling frequency: "))
m = int(input("Enter the filter order: "))
filter =
sig.firwin(m,pass_bands,pass_zero='highpass',window='hamming',fs=fs)
response = sig.freqz(filter,worN=8000)
plt.plot(fs*response[0]/2/np.pi,np.abs(response[1]),label='frequency
response')
plt.grid()
plt.title('Highpass Filter')
plt.xlabel('frequency')
plt.ylabel('gain')
plt.show()
```

Input:

```
Enter the pass band: 1500
Enter the stop band: 500
Enter the sampling frequency: 10000
Enter the filter order: 31
```

Output:



Assignment Question 3: Low Pass Filter

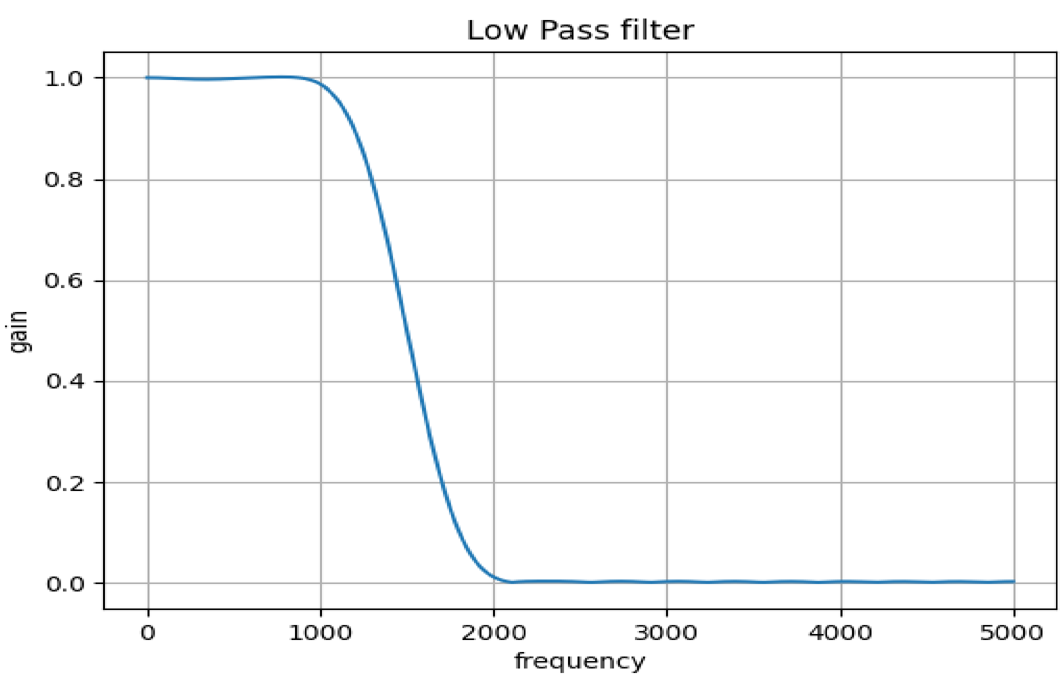
Code:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as sig
#parameters
pass_bands = int(input("Enter the pass band: "))
stop_bands = int(input("Enter the stop band: "))
fs = int(input("Enter the sampling frequency: "))
m = int(input("Enter the filter order: "))
filter =
sig.firwin(m,pass_bands,pass_zero='lowpass',window='hamming',fs=fs)
response = sig.freqz(filter,worN=8000)
plt.plot(fs*response[0]/2/np.pi,np.abs(response[1]),label='frequency
response')
plt.grid()
plt.title('Low Pass filter')
plt.xlabel('frequency')
plt.ylabel('gain')
plt.show()
```

Input:

```
Enter the pass band: 1500
Enter the stop band: 500
Enter the sampling frequency: 10000
Enter the filter order: 31
```

Output:



Assignment Question 4: Band Pass Filter

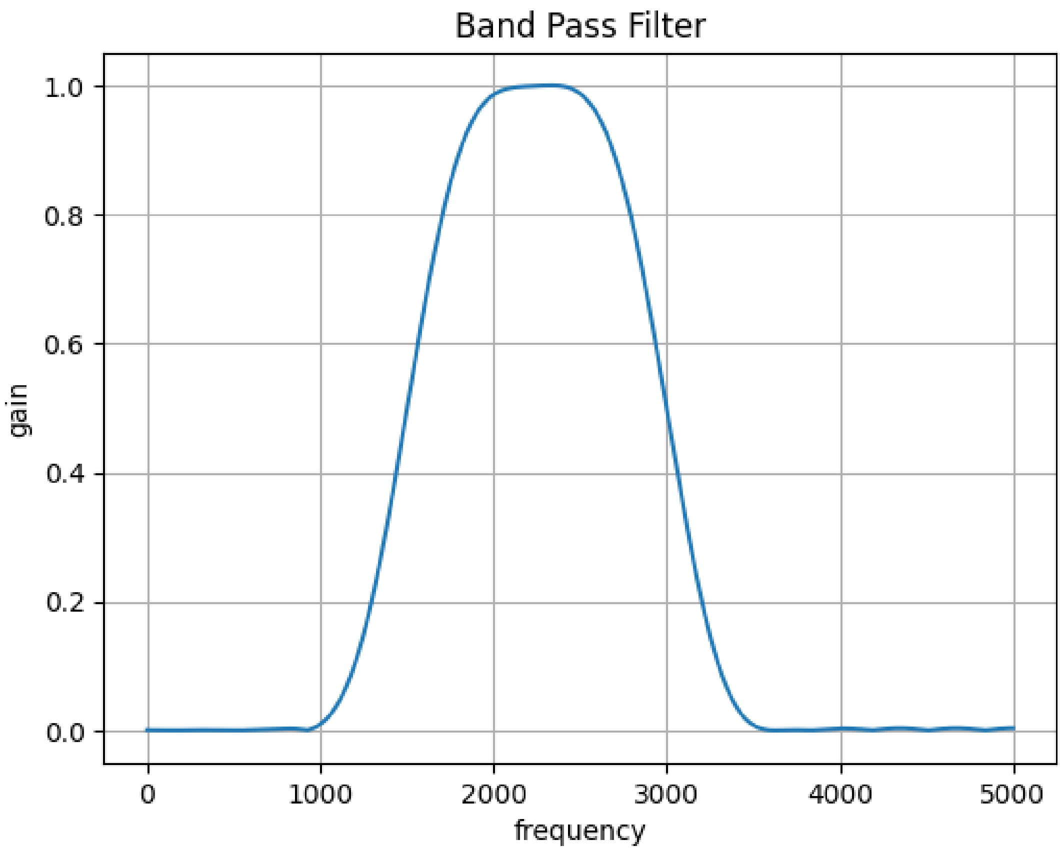
Code:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as sig

f1 = int(input("Enter the first pass band band: "))
f2 = int(input("Enter the second pass band: "))
stop_bands = int(input("Enter the second band: "))
fs = int(input("Enter the sampling frequency: "))
m = int(input("Enter the filter order: "))
#parameters
pass_bands = f1,f2

filter =
sig.firwin(m,pass_bands,pass_zero='bandpass',window='hamming',fs=fs)
response = sig.freqz(filter,worN=8000)
plt.plot(fs*response[0]/2/np.pi,np.abs(response[1]),label='frequency
response')
plt.grid()
plt.title('Band Pass Filter')
plt.xlabel('frequency')
plt.ylabel('gain')
plt.show()
```

Input:

```
Enter the first pass band band: 1500
Enter the second pass band: 3000
Enter the stop band: 500
Enter the sampling frequency: 10000
Enter the filter order: 31
```

Output:



Assignment Question 5: Band Stop Filter

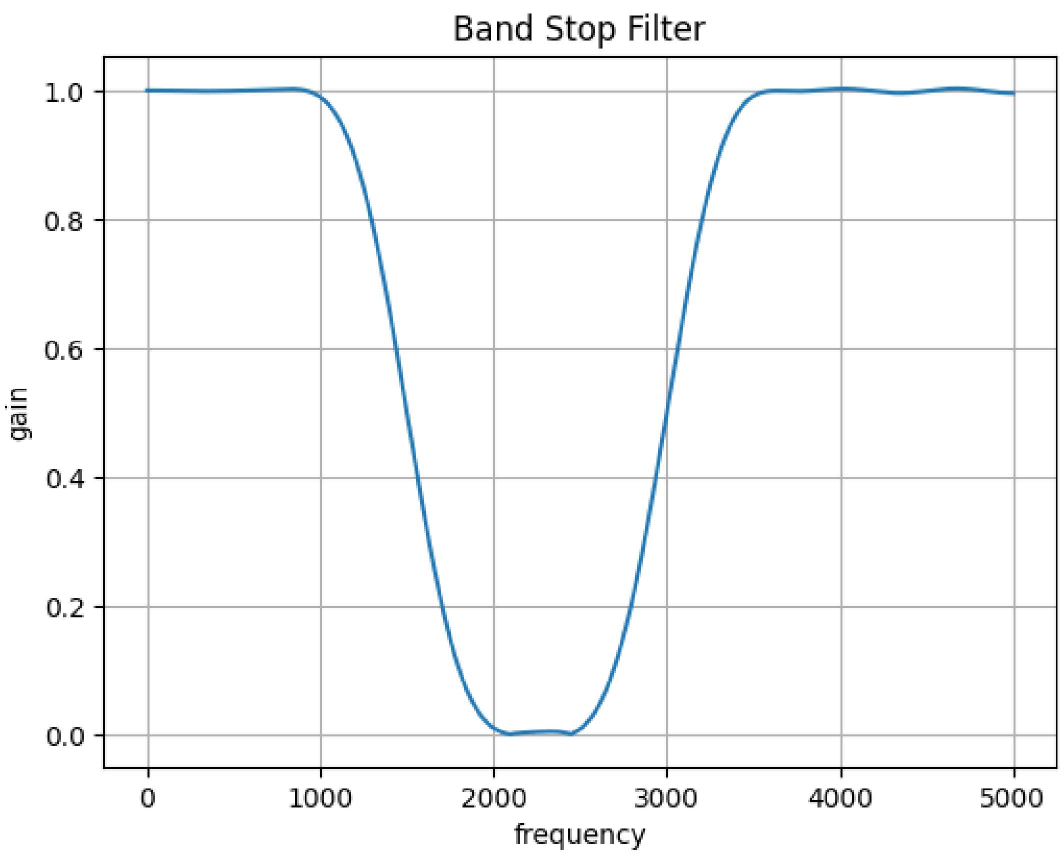
Code:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as sig

f1 = int(input("Enter the first pass band band: "))
f2 = int(input("Enter the second pass band: "))
stop_bands = int(input("Enter the second band: "))
fs = int(input("Enter the sampling frequency: "))
m = int(input("Enter the filter order: "))
#parameters
pass_bands = f1,f2

filter =
sig.firwin(m,pass_bands,pass_zero='bandstop',window='hamming',fs=fs)
response = sig.freqz(filter,worN=8000)

plt.plot(fs*response[0]/2/np.pi,np.abs(response[1]),label='frequency
response')
plt.grid()
plt.title('Band Stop Filter')
response')
plt.xlabel('frequency')
plt.ylabel('gain')
plt.show()
```

Input:

```
Enter the first pass band band: 1500
Enter the second pass band: 3000
Enter the stop band: 500
Enter the sampling frequency: 10000
Enter the filter order: 31
```

Output:



Assignment Question 6: Pass Band Filter

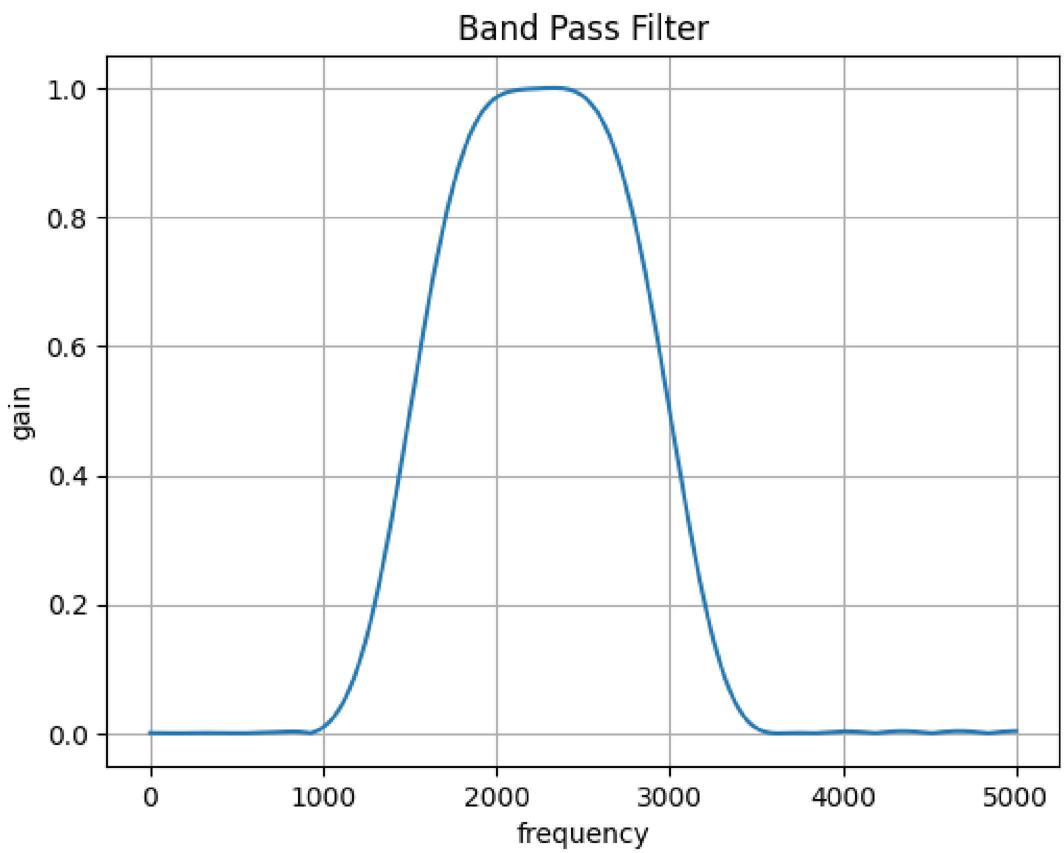
Code:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as sig

f1 = int(input("Enter the first pass band band: "))
f2 = int(input("Enter the second pass band: "))
stop_bands = int(input("Enter the second band: "))
fs = int(input("Enter the sampling frequency: "))
m = int(input("Enter the filter order: "))
#parameters
pass_bands = f1,f2

filter =
sig.firwin(m,pass_bands,pass_zero='bandpass',window='hamming',fs=fs)
response = sig.freqz(filter,worN=8000)
plt.plot(fs*response[0]/2/np.pi,np.abs(response[1]),label='frequency
response')
plt.grid()
plt.title('Band Pass Filter')
plt.xlabel('frequency')
plt.ylabel('gain')
plt.show()
```

Input:

```
Enter the first pass band band: 1500
Enter the second pass band: 3000
Enter the stop band: 500
Enter the sampling frequency: 10000
Enter the filter order: 31
```

Output:



Assignment Question 7: FIR Filter (Band Stop Filter)

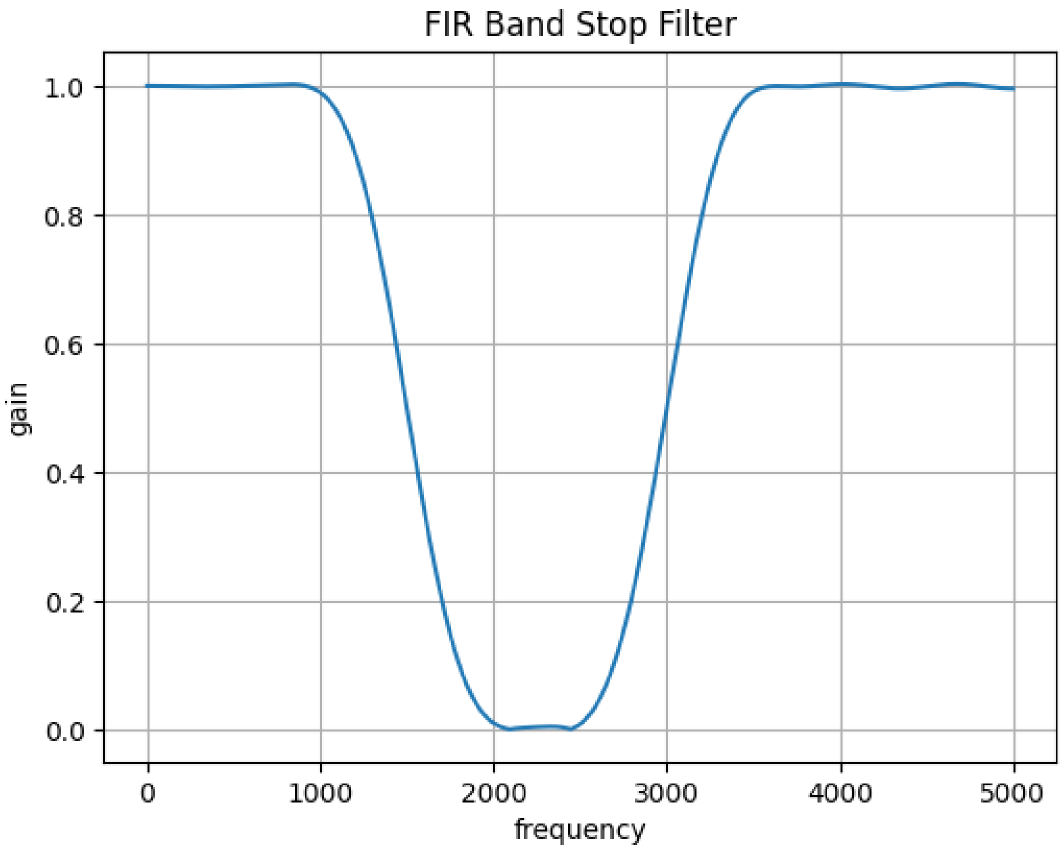
Code:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as sig

f1 = int(input("Enter the first pass band band: "))
f2 = int(input("Enter the second pass band: "))
stop_bands = int(input("Enter the second band: "))
fs = int(input("Enter the sampling frequency: "))
m = int(input("Enter the filter order: "))
#parameters
pass_bands = f1,f2

filter =
sig.firwin(m,pass_bands,pass_zero='bandstop',window='hamming',fs=fs)
response = sig.freqz(filter,worN=8000)

plt.plot(fs*response[0]/2/np.pi,np.abs(response[1]),label='frequency
response')
plt.grid()
plt.title('FIR Band Stop Filter')
plt.xlabel('frequency')
plt.ylabel('gain')
plt.show()
```

Input:

```
Enter the first pass band band: 1500
Enter the second pass band: 3000
Enter the stop band: 500
Enter the sampling frequency: 10000
Enter the filter order: 31
```

Output:



Assignment Question 8: IIR Filter

Code:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import butter,lfilter,freqz


fs = int(input("Enter the sampling rate in Hz: "))


time = np.arange(0,1,1/fs)
s=
np.sin(2*np.pi*5*time)+np.sin(2*np.pi*15*time)+np.sin(2*np.pi*30*tim
e)


def butter_stopband_filter(data,lowcut,highcut,fs,order=4):
    nyquist= 0.5*fs
    low = lowcut/nyquist
    high = highcut/nyquist
    b,a = butter(order,[low,high],btype='bandpass')
    y= lfilter(b,a,data )
    return y

plt.figure(figsize=(10,4))
plt.plot(time,s)
plt.title('orginal Signal')
plt.grid()
plt.show()

filtered_signal= butter_stopband_filter(s,10,25,fs )

plt.figure(figsize=(10,4))
plt.plot(time,filtered_signal)
plt.title('Filtered Signal')
plt.grid()
plt.show()
plt.show()
```

Input:

```
Enter the sampling rate in Hz: 100
```

Output: